TUTORIAL 2

NodeJS & OpenLayers & PostGIS

Abdülkadir Çakır

05/05/2020

***Implementation of Database***

Let’s create a table with these parameters:

**CREATE** **TABLE** public.trees (

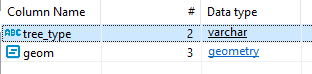
tree\_type **varchar** **NULL**,

geom geometry(POINT, 4326) **NULL**

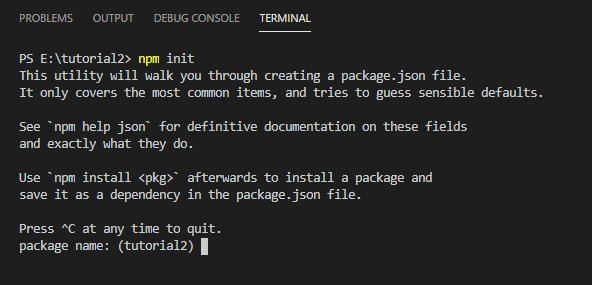
);

PS: Don’t forget to run :

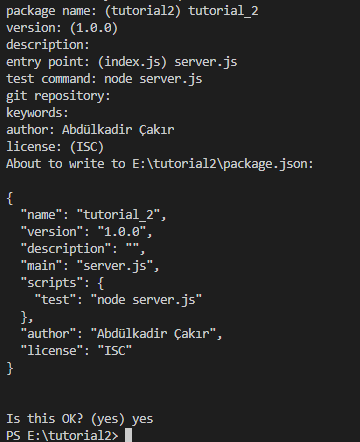
CREATE EXTENSION POSTGIS;



It is time to create Node JS project.



Fill it like this:



Install packages using npm install: (example: npm install pg)

pg

express

body-parser

Create a file named server.js (depends on what did you write on npm init entry point )

And go with these steps:

* Importing plugins
* Create DB Connection
* Enabling body-parser for POST method.
* Setting up port
* Setting up working directory

const express = require("express");

const pg=require("pg").Pool;

const bodyParser = require('body-parser');

const app = express();

const pool=new pg({host:'localhost',database:'hw\_db',user:'postgres',password:'postgres',port:'5432',ssl:false});

app.use(bodyParser.urlencoded({extended: true}));

app.use(bodyParser.json());

const \_port = process.env.PORT || 5000;

const \_app\_folder = \_\_dirname + '/dist' ;

app.use(express.static(\_\_dirname + '/dist' ));

After that you can implement you APIs.

* Implement data retrieval API (GET parameter will be enough)
* And data insert API (POST parameter will be enough)

var table\_name="trees";

app.get("/api/data",function(req,res)

{

    pool.query("SELECT jsonb\_build\_object('type','FeatureCollection','features', jsonb\_agg(feature)) FROM (SELECT jsonb\_build\_object('type','Feature','geometry',ST\_AsGeoJSON(geom)::jsonb, 'properties', to\_jsonb(row) - 'gid' - 'geom') AS feature  FROM (SELECT \* FROM "+table\_name+") row) features;", (err1, res1) =>

        {

            if(err1) {return console.log(err1);}

            res.json(res1.rows[0]["jsonb\_build\_object"]);

        });

});

app.post('/post', function(request, response){

    pool.query("INSERT INTO "+table\_name+" VALUES('"+request.body.tree\_type+"',ST\_SETSRID(ST\_MAKEPOINT("+request.body.Longitude+","+request.body.Latitude+"),4326));", (err1, res1) =>

        {

            if(err1)

                {   console.log(request.body);

                    return console.log(err1);}

                response.statusCode = 200;

                response.setHeader('Content-Type', 'text/plain');

                response.end('Data Store Success!\n');

        });

});

Now Implement express server startup codes.

app.all('\*', function (req, res) {

    res.status(200).sendFile(`/`, {root: \_app\_folder});

});

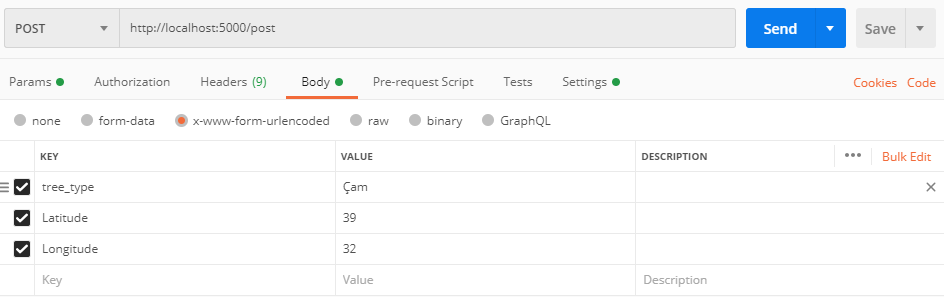
app.listen(\_port, function () {

    console.log("Node Express server for " + app.name + " listening on http://localhost:" + \_port);

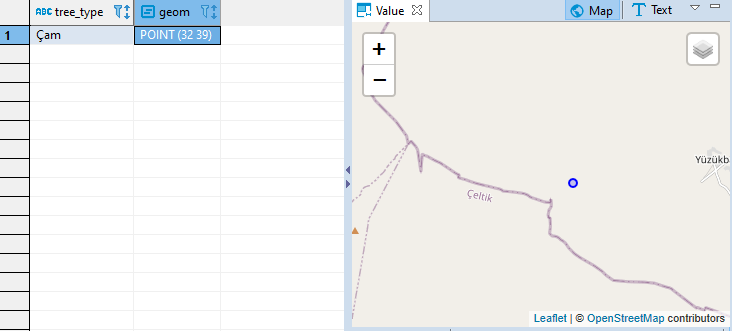
});

Run it using “node server.js”

You can use PostMan to check your APIs are working.

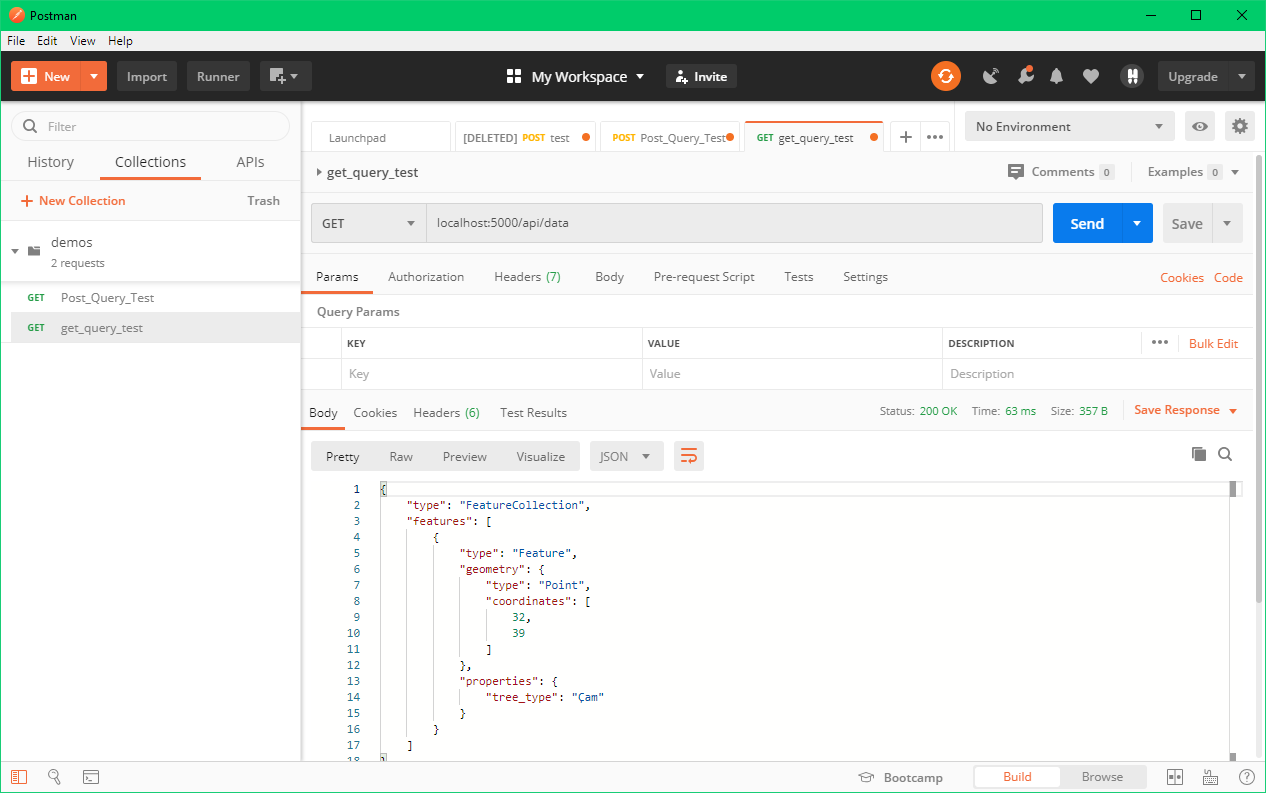


After pressing send button, Record inserted into table.



Let’s check GET API:

(No parameter required)



Last insert from POST request is available at both in database and /api/data request.

Let’s create front-end!

Create folder named with dist

And create new files in that folder:

* Index.html
* map\_handler.js
* styles.css

Index.html:

<!DOCTYPE html>

<html>

    <head>

        <link rel="stylesheet" href="https://cdn.rawgit.com/openlayers/openlayers.github.io/master/en/v6.2.0/css/ol.css">

        <link rel="stylesheet" href="styles.css">

        <script src="https://cdn.rawgit.com/openlayers/openlayers.github.io/master/en/v6.2.0/build/ol.js"></script>

    </head>

    <body>

        <div id="map" class="map"></div>

        <script src="map\_handler.js"></script>

        <div id="right\_panel" class="right\_panel">

            <br>

            <br>

            <br>

            <p>Tree Type:</p>

            <select id="tree\_type">

                <option value="Çam">Çam Ağacı</option>

                <option value="Kavak">Kavak Ağacı</option>

            </select>

            <p>Latitude:</p>

            <input type="text" id="Latitude"></input>

            <p>Longitude:</p>

            <input type="text" id="Longitude"></input>

            <br>

            <br>

            <br>

            <button onclick="submit();">Submit</button>

        </div>

    </body>

</html>

Map\_handler.js:

var baseMapLayer = new ol.layer.Tile({

    source: new ol.source.OSM()

});

var layer = new ol.layer.Tile({

  source: new ol.source.OSM()

});

var center = ol.proj.fromLonLat([32, 39]);

var view = new ol.View({

  center: center,

  zoom: 10

});

var map = new ol.Map({

    target: 'map',

    view: view,

    layers: [layer]

});

var vectorSource = new ol.source.Vector({

        url:"/api/data",

        format: new ol.format.GeoJSON({ featureProjection: "EPSG:4326" })

});

var stroke = new ol.style.Stroke({color: 'black', width: 2});

var fill = new ol.style.Fill({color: 'red'});

var markerVectorLayer = new ol.layer.Vector({

    source: vectorSource,

    style: new ol.style.Style({

        image: new ol.style.RegularShape({

          fill: fill,

          stroke: stroke,

          points: 4,

          radius: 10,

          angle: Math.PI / 4

        })

      })

});

map.addLayer(markerVectorLayer);

var select = new ol.interaction.Select({multiple:false});

select.on('select', fnHandler);

map.addInteraction(select);

map.on("click",handleMapClick);

function handleMapClick(evt)

{

  var coord=ol.proj.transform(evt.coordinate, 'EPSG:3857', 'EPSG:4326');

  document.getElementById("Latitude").value=coord[1];

  document.getElementById("Longitude").value=coord[0];

}

function fnHandler(e)

{

    var coord = e.mapBrowserEvent.coordinate;

    let features = e.target.getFeatures();

    features.forEach( (feature) => {

        console.log(feature.getProperties().tree\_type);

    document.getElementById("tree\_type").value=feature.getProperties().tree\_type;

    });

    if (e.selected[0])

    {

    var coords=ol.proj.transform(e.selected[0].getGeometry().getCoordinates(), 'EPSG:3857', 'EPSG:4326');

    document.getElementById("Latitude").value=coords[1];

    document.getElementById("Longitude").value=coords[0];

    console.log(coords);

    }

}

function submit()

{

    var xhr = new XMLHttpRequest();

    xhr.open("POST", "/post", true);

    xhr.setRequestHeader('Content-Type', 'application/json');

    var data=JSON.stringify({

        Latitude: document.getElementById('Latitude').value,

        Longitude: document.getElementById('Longitude').value,

        tree\_type: document.getElementById('tree\_type').value

    });

    xhr.send(data);

    location.reload();

}

Styles.css:

html,body

{

width: 100%;

height: 100%;

}

.map

{

    width: 80%;

    height: 100%;

    float: right;

}

.right\_panel

{

    background-color: rgb(94, 173, 226);

    width: 20%;

    height: 100%;

}

It’s ready!

Save and re-run the server.js

